IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

BRAMHILL et al.

Atty. Ref.: 36-1230

Serial No.: 09/091,735

Group Art Unit: 3625

Filed: June 24, 1998

Examiner: Nguyen, C.

For: COPY PROTECTION OF DATA

December 1, 2003
[November 30, 2003 = Sunday]

APPEAL BRIEF

Commissioner for Patents
Arlington, VA 22202

Sir:

Appellant hereby appeals the "final" rejection of August 8, 2003.[1]

REAL PARTY IN INTEREST

The real party in interest is the assignee, British Telecommunications public limited company, a corporation of Great Britain.

[1] A formal final rejection was mailed on July 30, 2003. On August 7, 2003, Appellant's representative telephoned the Examiner and argued that the finality of the Office Action was premature. In particular, the non-final Office Action mailed January 2, 2003 rejected claims 1-2, 5-6, 12, 28 and 30-33 under 35 U.S.C. §102 as allegedly being anticipated by Spies et al. No amendment to any of these claims was presented in the Response filed by Appellant on May 2, 2003. Nevertheless, claims 1-2, 5-6, 12, 28 and 30-33 were rejected under 35 U.S.C. §103 as allegedly being "obvious" over Spies et al. in view of Rhoads in the "final" rejection mailed on July 30, 2003. The grounds of rejection of claims 1-2, 5-6, 12, 28 and 30-33 therefore changed without any intervening amendment to these claims. The "final" Office Action faxed August 8, 2003 (which is being appealed in this Appeal Brief) allegedly resolves this issue by indicating that claims 1-8, 12, 14, 28 and 30-33 are rejected under 35 U.S.C. §103 over Spies et al. alone (see page 3, lines 1-2 of the "final" rejection faxed August 8, 2003). Appellant still submits that the grounds of rejection of claims 1-8, 12, 14, 28 and 30-33 were changed without an intervening amendment and thus submits that the finality of the August 8, 2003 Office Action is premature.

RELATED APPEALS AND INTERFERENCES

The Appellant, the undersigned, and the assignee are not aware of any related appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

STATUS OF THE CLAIMS

Claims 1-8, 12, 14-18, 21 and 28-38 remain pending in this application. Claims 1-8, 12, 14-18, 21 and 28-38 stand rejected by the Examiner, the rejections of which are appealed.

Non-elected claims 22-26 have been canceled. Claims 22-26 are being pursued in a divisional patent application.

STATUS OF ANY AMENDMENT FILED SUBSEQUENT TO FINAL REJECTION

No amendment to the claims has been filed subsequent to the Final Rejection of August 8, 2003. The claims as presented in the Appendix to this brief are as amended by the Amendments filed on May 2, 2003, August 28, 2002, November 16, 2001, April 24, 2001, June 20, 2000 and July 24, 1998.

CONCISE EXPLANATION OF THE INVENTION

The present invention relates to copy protecting data transmitted from a server to a client. An exemplary embodiment of the present invention is described below.

A webpage 7 containing copyright protected image data is downloaded from server 1 to client computer 3. Client 3 first uploads a request to server 1 for details of a webpage. The request to server 1 typically comprises a conventional hypertext file transfer protocol (HTTP) page request. Server 1 obtains or constructs the webpage and downloads the HTML code corresponding to the webpage to client computer 3. The downloaded HTML code may include a Java applet A1. Applet A1 is executed on client computer 3 using a Java interpreter within the webpage browser in order to prepare the browser to receive data to be displayed in, for example, a copyright protected region 12 of webpage 7.

The execution of applet A1 causes a request for a file of copyright protected data to be uploaded to server 1. Server 1 then performs an authentication step in order to determine whether it is safe to download the copyright protected data file requested by client computer 3. Assuming that client computer 3 passes this authentication step, server 1 prepares and transmits the copyrighted protected data file to client computer 3.

Preparing the copyright protected data file for downloading to client computer 3 involves watermarking the data to be downloaded. Watermarking provides additional security in the event the protected data is copied because knowledge of the source of copying can be determined from the watermark. (See step 10.2 in Fig. 6.) The watermarked data is hashed at server 1 using a copy of a hashing algorithm HA that was downloaded in applet A1 and a file specific session hashing key K_{SH}. This hashing ensures that sections of data are not removed and replaced by data such as to ensure that for example a command "pay $100" is not changed to "pay $1.00." (See

step S10.3.) The data is then encrypted at server 1 using a copy of an encryption algorithm EA and a key K_E which are downloaded previously to client computer 3 in the Java byte codes of applet A1. The encryption algorithm EA forms a pair of algorithms, one of which is to encrypt and the other is to decrypt. (See step S10.4.) The resulting file is then wrapped in a proprietary file format which includes additional cryptographic protection techniques. (See step S10.5.) The resulting file of copyright protected data is then downloaded to client computer 3. (See step S11.)

The downloaded data file is then processed using applet A1 which was previously downloaded to client computer 3. Applet A1 allows the downloaded data file to be decrypted and checked for integrity (i.e., hash verified). Specifically, if the integrity of the content of a header is satisfactorily verified (see step S12.1), applet A1 checks whether it knows how to process files of the type specified in the header (see step S12.2). If the result of the check is satisfactory, applet A1 can make use of the specific copyright protected control information for the file present in header H when processing user requests for data manipulation. Specifically, the downloaded data file is decrypted using the encryption algorithm EA and the key K_E previously downloaded in applet A1. The integrity of the decrypted file is then verified. (See steps S12.6-12.7.) If this integrity check is satisfactory, applet A1 can display the content of the decrypted file in region 12 (See Fig. 4.) Accordingly, if all of the integrity checks are satisfied, the decrypted data may be copied or saved.

If the integrity check of the content header H is unsatisfactory, applet A1 determines that it does not know how to process files of the type specified in the header, or if the integrity of the decrypted file is unsatisfactory, an error banner is

displayed in region 12 (Fig. 4). (See step S12.3). If the downloaded copyright protected data file contains image data, the image is displayed, together with its imperceptible watermark in region 12. The user, however, cannot save or copy the image data. Because the Java enabled browser is executing an applet for the image data in region 12, the functions of a right mouse button including print, save or copy are disabled for region 12. Therefore, if a user clicks a right button of the mouse for region 12, no menu option is automatically provided for saving, copying or printing the displayed data in region 12 in order to prevent unauthorized copying.

When a user operating client computer 3 requests access to data, such as a particular webpage, instead of server 1 just sending the requested data to client computer 3, another program (an applet) is triggered to run at client computer 3 and take over the process for obtaining the requested data from server 1 and providing a controlled environment at client computer 3 in which the decrypted data may be viewed by the user. Since all of this executes in the background, the user will not normally be aware that anything else is going on beyond server 1 merely downloading the data as requested, unless the user tries to perform copyright-infringing actions at client computer 3.

CONCISE EXPLANATION OF THE ISSUES PRESENTED FOR REVIEW

Whether claims 1-2, 5-8, 12, 14, 21 and 28-38 are made "obvious" under 35 U.S.C. §103 based on Spies et al. (US '314, hereinafter "Spies").[2] Whether claims 3-4 and 16 are made "obvious" under 35 U.S.C. §103 based on Spies in view of Rhoads

[2] See page 3, lines 1-2 of the Final Office Action.

(US '978).[3] Whether claim 15 is made "obvious" under 35 U.S.C. §103 based on Spies in view of Probst (US '899).[4] Whether claim 17 is made "obvious" under 35 U.S.C. §103 based on Spies in view of Official Notice.[5] Whether claim 18 is made "obvious" under 35 U.S.C. §103 based on Spies in view of Crawford (US '651).[6]

WHETHER THE CLAIMS STAND OR FALL TOGETHER

Claims 1-8, 12, 14-18, 21, 28-29, 31-32, 34-35 and 37-38 stand or fall together and do not stand or fall with any other claims.

Claims 30, 33 and 36 stand or fall together and do not stand or fall with any other claims.

The specific reasons for each of the above groups of claim(s) standing or falling together or alone is provided below in the section entitled "Arguments with Respect to the Issues Presented for Review."

ARGUMENTS WITH RESPECT TO THE ISSUES PRESENTED FOR REVIEW

Claims 1-2, 5-8, 12, 14, 21 and 28-38 are not "obvious" under 35 U.S.C. §103 over Spies.

In order to establish a prima facie case of obviousness, all of the claimed limitations must be taught or suggested by the prior art. Appellant respectfully submits that Spies fails to teach or suggest all of the claimed limitations. For example, Spies fails to teach or suggest selectively controlling access to copy or save

[3] See page 7, lines 13-15 of the Final Office Action.
[4] See page 9, lines 1-3 of the Final Office Action.
[5] See page 10, lines 5-6 of the Final Office Action.
[6] See page 10, lines 12-14 of the Final Office Action.

functions at the client in respect of data in its unprotected form, as required by independent claims 1, 28-29 and 35. Independent claims 30-34 and 36 require similar limitations. Spies fails to teach or suggest restricting or preventing access to copy or save functions of data in its unprotected form as required by independent claims 31-33 and 36 or suppressing client computer copy or save functions with respect to an unprotected copy of the requested data as required by independent claims 30 and 34.

Spies is directed to a method of cryptographically protecting video content using cryptographic keys to enable secure ordering and transferring of video content from a server (e.g., a video content provider computer) to a client (e.g., a user's set top box). Spies is particularly concerned with ensuring that if video content is intercepted in transit between a server and a client, or when in possession of user (e.g., when stored on a DVD or upon arrival at a user's set-top box or other computer unit), pirate copies cannot be made. This is achieved by Spies through a particular method of storing and exchanging cryptographic keys so that when the video content is accessible, it is encrypted in a way that would be almost impossible to decrypt without access to the appropriate keys. In particular, Spies discloses an integrated circuit (IC) card arranged to interface with a user's set-top box to store the essential decryption capabilities (see, e.g., col. 8, lines 26-43) with respect to particular video content ordered by the user. Those capabilities are downloaded to the IC card once the order has been validated. The IC card may also be arranged to implement at least a part of a decryption program stored in the IC card to decrypt downloaded encrypted packets so that the ordered video content may displayed on the user's display device (e.g., a television).

While Spies discusses preventing unauthorized access to cryptographic keys and ensuring that video content remains in an encrypted form when outside the user's set-top box, Spies fails to teach or suggest any measures to selectively control, restrict, prevent or suppress access to video content within the set-top box. That is, Spies fails to teach or suggest selectively controlling, restricting, preventing or suppressing access to copy or save functions once the data has been decrypted and is in a form that can be sent to a display device for display. Unauthorized access to already decrypted data within the set-top box is not considered at all by Spies to be a problem.

With respect to the claimed feature of restricting or preventing access to copy or save functions of data in its unprotected form, the Final Office Action states "The examiner submits although Spies may not expressly disclose these (sic) claim language, Spies inherently includes this function." (See Section 5 of the Final Office Action.) In particular, the Final Office Action apparently alleges that column 1, lines 45-49 inherently (but not explicitly) discloses the above claimed features. (See page 1, lines 15-17 and page 3, lines 12-15 of the Final Office Action). Column 1, lines 45-49 of Spies states "On the other hand, when the video is distributed on **cassette or DVD**, the viewer is considered to have the ability to record and redistribute the video with little difficulty (emphasis added)." Appellant disagrees with the Examiner's allegation that the claimed limitation of restricting or preventing (or selectively controlling or suppressing) access to copy or save functions of data in its unprotected form is inherent from column 1, lines 45-49 of Spies.

First, the sentence provided by column 1, lines 45-49 is directed to a situation in which video is distributed on cassette (i.e., videocassette) or DVD. The substance

of this sentence is therefore not even applicable to the client-server systems required by independent claims 1, 30, 31 and 33-36 or the server computers required by independent claims 28, 29 and 32. The Examiner's reliance on column 1, lines 45-49 is completely misplaced since the claimed invention is directed to a client-server environment or a server computer environment, not to a videocassette or a DVD.

Even if the sentence of column 1, lines 45-49 were applicable to the claimed invention (Appellant submits that it is not for the reasons discussed in the preceding paragraph), Appellant submits that the claimed limitation of restricting or preventing access to copy or save function of data in its unprotected form is not inherent from Spies's teaching that "the viewer is considered to have the ability to record and redistribute the video with **little difficulty** (emphasis added)". The Examiner argues that "Although Spies mentions there is 'little difficulty', the Examiner submits that a level difficulty does exist..." (See Section 4, lines 2-3). Appellant submits that the Examiner has therefore misinterpreted "little difficulty" to mean something virtually opposite to its conventional interpretation. Indeed, this portion of Spies indicates that users can record and redistribute the video stored on a videocassette or DVD with virtually **no** difficulty. In marked contrast, if access to copy or save functions is restricted or prevented as claimed, an unauthorized user would experience a **high** degree of difficulty in copying or saving the data.

Moreover, the access to copy or save functions in its unprotected form is controlled by execution of a program portion. In marked contrast, Spies fails to teach or suggest that the "little difficulty" to record and redistribute video originating from a videocassette or DVD is provided by execution of a program portion. For example,

the "little difficulty" may result, for example, from the little effort that it takes the user to press the record button on a video recorder. Any amount of "difficulty" in the "little difficulty" mentioned by Spies to record and redistribute video on a videocassette or DVD therefore has nothing to do with execution of a program portion as claimed.

Claim 1 further requires running a program portion at a client to generate and upload to the server a request for access to data. Claim 1 also requires that this same program portion convert the cryptographically protected data to an unprotected form and selectively control access to copy or save functions of the data in its unprotected form. Claim 1 therefore requires that the same program portion generates and uploads a request for access to data and performs conversion of the cryptographically protected data to an unprotected form. The "same" program portion forms a set of instructions forming a single entity, defined independently of the processing environment in which the instructions might eventually be executed (although when they are executed, it would be in the context of the instructions being part of a single entity). The same program portion may be downloaded (see, e.g., dependent claim 7) as a whole, not fragmented and executed piecemeal. The "same" program portion allows an unbroken chain of control from the time during which the data set is in a protected form (i.e. an encrypted form) to control during conversion (e.g., decryption) of the data set into an unprotected form and for as long as the data set remains in the unprotected form. The chain of control provided by the "same" program portion thus remains unbroken. Independent claims 28, 29, 31, 32 and 35 also require that the

same program portion generates a request for access to data and performs conversion of the cryptographically protected data to an unprotected form.

In contrast, Spies discloses a video purchasing application program or a video on demand (VOD) application program which runs at a user's "viewing computing unit" (see col. 13, line 24 *et seq.* or col. 15, line 18 *et seq.*) to enable a user to select a video to be downloaded and to generate an order for the selected video. Spies then discloses performing order validation and key encryption and a downloading process to encrypt and download the ordered video content. However, once the encrypted video content is delivered to the client (e.g., the user's set-top box), Spies discloses using a separate program - a "decryption routine 162" in the embodiment shown in Fig. 7 or a separate "decryption unit 170" in the embodiment illustrated in Fig. 8 - to control perform the decryption of the downloaded data. The decryption routine or decryption unit is therefore not the same program portion or unit as that which (e.g., video purchasing application program) generates and uploads the request (order) for access to video data.

Claims 3-4 and 16 are not "obvious" under 35 U.S.C. §103 over Spies in view of Rhoads (U.S. '978). Claim 15 is not "obvious" under 35 U.S.C. §103 over Spies in view of Probst (U.S. '899). Claim 17 is not "obvious" under 35 U.S.C. §103 over Spies in view of Official Notice. Claim 18 is not "obvious" under 35 U.S.C. §103 over Spies in view of Crawford (U.S. '651).

Claims 3-4 and 15-18 depend from independent claim 1. All of the above comments with respect to Spies and claim 1 therefore apply equally to these

dependent claims. Neither Rhoads, Probst, Official Notice nor Crawford remedies the above deficiencies of Spies with respect to the invention of claim 1.

CONCLUSION

For all of the reasons set forth above, it is respectfully requested that this appeal be granted and that the rejections discussed above be reversed.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: Raymond Nich 4/4/26
for Larry S. Nixon
Reg. No. 25,640

LSN:RYM:ap
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703)816-4000
Facsimile: (703)816-4100

APPENDIX OF CLAIMS ON APPEAL

1. A method of protecting data sent from a server to a client, said method comprising:

running a program portion at the client, the program portion generating and uploading to the server a request for access to data;

cryptographically protecting the data;

sending the cryptographically protected data to the client; and

after the running of the program portion has begun and under control of the program portion at the client, converting the cryptographically protected data to an unprotected form and selectively controlling access to copy or save functions at the client in respect of the data in its unprotected form.

2. A method as in claim 1 wherein cryptographically protecting the data comprises protecting the data by encryption.

3. A method as in claim 1 wherein cryptographically protecting the data comprises protecting the integrity of the data cryptographically.

4. A method as in claim 3 wherein the integrity of the data is achieved by hashing.

5. A method as in claim 1 including authenticating that the client is permitted to receive the data.

6. A method as in claim 1 including identifying the client to the server before the data are sent to the client.

7. A method as in claim 1 including:

generating the program portion at a server,

downloading the program portion to the client, and

running the program portion on the client such that a request is uploaded to the server for a file containing the cryptographically protected data.

8. A method as in claim 7 wherein the program portion is generated in response to, and corresponds with, an earlier received request for access to the data.

12. A method as in claim 1 wherein the data are sent to the client from the server through a network.

14. A method as in claim 7 wherein the program portion includes data concerning a cryptographic key, and the method including using the key to render the downloaded cryptographically protected data into an unprotected form.

15. A method as in claim 1 wherein the server and the client each hold data corresponding to a cryptographic key and a machine identifier for uniquely identifying the client, the method including:

sending a challenge to the client, such that it generates a signed response as a cryptographic function of the key and the machine identifier held therein,

generating from the cryptographic key and machine identifier held associated with the server, a corresponding signed response as a cryptographic function of the key and the machine identifier,

comparing the signed responses from the client and the server, and if they correspond, performing the cryptographic protection of the data with the key, and

converting the cryptographically protected data into an unprotected form at the client with the key.

16. A method as in claim 1 wherein the data is steganographically marked.

17. A method as in claim 1 including registering the client with the server.

18. A method as in claim 1 including:

determining a machine identifier of the client by analysing its hardware and/or its software configuration,

transmitting the machine identifier to the server,

combining the transmitted machine identifier with a cryptographic key to form a unique determinator for the client,

transmitting the unique determinator to the client, to be stored therein for use subsequently in identifying the client to the server, to permit encypted data to be downloaded thereto from the server.

21. A data storage medium storing copy protected data on the client received by a method according to claim 1.

28. A server for providing access to data sets in a protected form, the server comprising:

an input for receiving a request for access to a data set;

protecting means for cryptographically protecting the requested data set; and

generating means for generating a program portion for sending to the source of the access request,

wherein said program portion is operable and after the program portion is permitted to run at the source of the access request, in use:

to generate a request for access to the cryptographically protected data set;

on receipt of the cryptographically protected data set, to convert it into an unprotected form; and

to selectively control access to copy or save functions in respect of the data set when in said unprotected form.

29. A computer program carrier medium containing a computer program which are executable by a computer to perform method steps for implementing a server, the method steps comprising:

receiving a request for access to a data set;

cryptographically protecting the requested data set; and

generating a program portion for sending to the source of the access request,

wherein said program portion is operable and after the program portion is permitted to run at the source of the access request, in use:

generating a request for access to the cryptographically protected data set;

on receipt of the cryptographically protected data set, converting it into an unprotected form; and

selectively controlling access to copy or save functions in respect of the data set when in said unprotected form..

30. A method of protecting data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and

running a program at the client so that after running the program at the client has begun at the client, the program serves to both: (a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) suppress client computer copy and save functions with respect to the unprotected copy of the requested data.

31. A method of protecting data sent from a server to a client, said method comprising:

running a program portion at the client, the program portion generating and uploading to the server a request for access to data;

cryptographically protecting the data;

sending the cryptographically protected data to the client; and

after access to the program portion is permitted and under control of the program portion, converting the cryptographically protected data to an unprotected form and restricting or preventing access to copy or save functions at the client in respect of the data in its unprotected form.

32. A server for providing access to data sets in a protected form, the server comprising:

an input for receiving a request for access to a data set;

protecting means for cryptographically protecting the requested data set; and

generating means for generating a program portion for sending to the source of the access request,

wherein after access to the program portion is permitted and said program portion is operable, in use:

to generate a request for access to the cryptographically protected data set;

on receipt of the cryptographically protected data set, to convert it into an unprotected form; and

to restrict or prevent access to copy or save functions in respect of the data set when in said unprotected form.

33. A method of protecting data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and

running a program at the client after access to the program is permitted to both: (a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) restrict or prevent client computer copy and save functions with respect to the unprotected copy of the requested data.

34. A method of controlling access to data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and

running a program at the client so that after running the program at the client has begun at the client, the program serves to both: (a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) suppress client computer copy or save functions with respect to the unprotected copy of the requested data.

35. A method of controlling access to data sent from a server to a client, said method comprising:

running a program portion at the client, the program portion generating and uploading to the server a request for access to data;

cryptographically protecting the data;

sending the cryptographically protected data to the client; and

after access to the program portion is permitted and under control of the program portion, converting the cryptographically protected data to an unprotected form and restricting or preventing access to copy or save functions at the client in respect of the data in its unprotected form.

36. A method of controlling access to data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and

running a program at the client after access to the program is permitted to both: (a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) restrict or prevent client computer copy or save functions with respect to the unprotected copy of the requested data.

37. A method as in claim 30, wherein the program running at the client generates and uploads a request for data from the client to the server, and the protected copy of requested data is downloaded from the server to the client in response to the request.

38. A method as in claim 33, wherein the program running at the client generates and uploads a request for data from the client to the server, and the protected copy of requested data is downloaded from the server to the client in response to the request.